

Preguntas para Reunión MuleSoft - Lunes

Proyecto: Signature Router – Integración con MuleSoft API Gateway

Fecha Reunión: Lunes [FECHA]

Participantes: DevOps Team / MuleSoft Team

Objetivo: Obtener especificaciones técnicas completas para Epic 11

RESUMEN EJECUTIVO - PUNTOS CRÍTICOS DE LA REUNIÓN

TOP 3 PREGUNTAS MÁS IMPORTANTES (preguntar PRIMERO)

1. CONFIRMADO: Canales Disponibles - Alcance Definitivo

-  SMS – Disponible → **IMPLEMENTAR**
-  PUSH – Disponible → **IMPLEMENTAR**
-  EMAIL – Disponible (bonus, no planeado originalmente)
-  VOICE – NO disponible → **FUERA DE ALCANCE** (no se implementará)
-  BIOMETRIC – NO disponible → **FUERA DE ALCANCE** (no se implementará)

ALCANCE FINAL Epic 11: Solo SMS + PUSH (VOICE y BIOMETRIC quedan fuera)

2. CRÍTICO: ¿Qué es el campo "practice": "monkey"?

- Visto en ejemplo de request SMS
- ¿Es obligatorio? ¿Valores posibles?
- ¿Afecta al routing o procesamiento?

3. ¿Los interfaces pueden incluir metadata del provider real? (NUEVO - CRÍTICO)

- **CONFIRMADO:** Response actual NO incluye metadata de providers
- Necesitamos saber qué provider usó MuleSoft (Twilio, Firebase, etc.)
- Necesitamos latencia específica del provider (sin overhead de MuleSoft)
- Necesitamos errores específicos del provider para troubleshooting
- **Solicitamos formalmente ampliación de interfaces** (ver sección 9.1)

4. ¿Cuándo podemos empezar a integrar? (timeline, credenciales, sandbox)

- Necesitamos fecha concreta de inicio
- Client ID + Client Secret para DEV/UAT/PROD
- Acceso a ambiente de desarrollo/sandbox

Impacto de NO tener Metadata de Providers

Funcionalidad Afectada	Impacto	Severidad
Dashboard de performance por provider	Pérdida total de visibilidad	● CRÍTICO
Troubleshooting de errores	10x más lento (escalaciones a MuleSoft)	● CRÍTICO
Cumplimiento de SLAs (P99 < 500ms)	No podemos medir/optimizar	● CRÍTICO
Optimización de costos	Ahorro potencial 40% perdido	● ALTO
A/B testing de providers	Imposible hacer testing	● ALTO
Alertas específicas por provider	Solo alertas genéricas	● ALTO
Auditoría y compliance	Trazabilidad insuficiente	● ALTO

Conclusión: Sin metadata, el sistema funciona pero **pierde 80% de su valor analítico y observabilidad.**

Checklist de Preparación PRE-Reunión

Antes de la reunión, asegúrate de tener:

- Este documento impreso o en segunda pantalla
- Laptop con acceso a:
 - Postman (para probar endpoints en vivo)
 - Terminal/PowerShell (para curl commands)
 - Editor de texto (para copiar specs)
- Ejemplos concretos preparados:
 - Ejemplo de request SMS que quieras enviar
 - Ejemplo de dashboard que necesitas poblar
 - Ejemplo de error que necesitas debuggear
- Contactos clave:
 - Email del contacto técnico de MuleSoft (para seguimiento)
 - Slack/Teams channel para dudas rápidas

🎤 Estrategia de Comunicación en la Reunión

1. **Primeros 5 minutos:** Contextualizar proyecto Signature Router
2. **Siguientes 10 minutos:** Preguntar por canales disponibles (**pregunta crítica #1**)
3. **Siguientes 15 minutos:** Discutir metadata de providers (**pregunta crítica #2**)
 - Mostrar ejemplo de dashboard que necesitas
 - Explicar impacto de no tenerlo (usar tabla de impacto arriba)
 - **Enfatizar:** "Los interfaces NO son inmutables, podemos solicitar ampliación"
4. **Siguientes 20 minutos:** Detalles técnicos (auth, endpoints, errores)
5. **Últimos 10 minutos:** Timeline, entregables, próximos pasos

Frase clave para metadata:

"Entendemos que MuleSoft gestiona la complejidad de providers, pero necesitamos visibilidad para cumplir nuestros SLAs con clientes. ¿Los interfaces actuales pueden incluir metadata del provider real, o podemos solicitar que se amplíen?"

⌚ Contexto Rápido

El **Signature Router** debe integrarse **obligatoriamente** con **MuleSoft API Gateway** como **única** capa de comunicación con providers externos (normativa corporativa).

API CONFIRMADA: Singular Bank - Communication Services API (v1)

- **API ID:** 2611145
- **Asset Version:** 1.0.3
- **Implementation URI:** <https://api.selfbank.es/system/commu...> (verificar URL completa)
- **Status:** Active
- **Mule Version:** 4.10.0

Canales potenciales:

-  **SMS** (actualmente Twilio)
-  **Voice/Call** (actualmente Twilio Voice)
-  **Push Notifications** (actualmente Firebase FCM)
-  **Biometric** (futuro)

⚠ IMPORTANTE: Si MuleSoft NO tiene un canal implementado (ej: Biometric), ese canal **NO estará disponible** en Signature Router. **NO se permite** comunicación directa con providers (normativa).

⚠ PREGUNTA MÁS CRÍTICA (PREGUNTAR PRIMERO)

🚧 Cobertura de Canales Disponibles

¿Qué canales de notificación tienen YA implementados en MuleSoft?

Esta es la pregunta **MÁS IMPORTANTE** porque determina qué funcionalidades tendrá Signature Router.

- SMS:** ¿Tienen endpoint implementado? → POST /api/v1/???
 - Si **SÍ** → Migrar de Twilio directo a MuleSoft
 - Si **NO** → SMS **NO estará disponible** en Signature Router ✗
- PUSH Notifications:** ¿Tienen endpoint implementado? → POST /api/v1/???
 - Si **SÍ** → Migrar de Firebase directo a MuleSoft
 - Si **NO** → PUSH **NO estará disponible** en Signature Router ✗
- VOICE/Call:** ¿Tienen endpoint implementado? → POST /api/v1/???
 - Si **SÍ** → Migrar de Twilio Voice directo a MuleSoft
 - Si **NO** → VOICE **NO estará disponible** en Signature Router ✗
- BIOMETRIC:** ¿Tienen endpoint implementado? → POST /api/v1/???
 - Si **SÍ** → Implementar desde cero vía MuleSoft
 - Si **NO** → BIOMETRIC **NO estará disponible** en Signature Router ✗

Canales NO disponibles:

- ¿Están en el **roadmap**? ¿Cuándo estarán listos?
- Si **NO están planificados**, ¿podemos solicitar su implementación?
- ¿Cuál es el **effort** estimado para implementar un canal nuevo?

Estrategia de implementación:

- ¿Puedo implementar **canal por canal** incrementalmente? (SMS primero, luego PUSH, etc.)
- ¿O debo esperar a que **todos** estén listos?
- ¿Qué **timeline** recomiendan para go-live?

HALLAZGOS DE LA DOCUMENTACIÓN (PDF Exchange)

Información Confirmada:

API Identificada:

- Nombre: Singular Bank - Communication Services API
- Versión: v1 (Asset version: 1.0.16)
- Tipo: REST API (RAML 1.0)
- Estado: Stable
- Conformance: Not Validated 
- Owner: Borja Esteban
- Engagement score: 0.15%

Canales Disponibles:

-  SMS - POST /communication-execution/sms-notification/execute
-  PUSH - POST /communication-execution/push-notification/execute
-  EMAIL - POST /communication-execution/email-notification/execute
-  VOICE - No mencionado
-  BIOMETRIC - No mencionado

Monitoreo:

-  Health Check - GET /health/retrieve
-  Métricas - GET /metrics/retrieve

Arquitectura BIAN v12.0:

- Basado en "Communication Execution" domain
- Integración con: Customer Communications, Customer Authentication
- Pattern: Ejecución inmediata (síncrona)

Puntos CRÍTICOS Detectados:

1. Campo misterioso "practice": "monkey" 
 - Aparece en ejemplo de request SMS
 - No hay explicación en documentación
 - **PREGUNTAR QUÉ ES Y SI ES OBLIGATORIO**

2. NO hay metadata de providers en responses

- Response solo incluye: notificationId, status, submittedAt, channel
- **NO incluye:** provider real, latencia, coste, fallback usado
- **SOLICITAR ampliación de interfaces (sección 9.1)**

3. "Not Validated" conformance status

- La API NO está validando conformidad con RAML
- Posibles diferencias entre spec y realidad
- **PREGUNTAR sobre discrepancias conocidas**

Response Actual vs. Response Necesario:

Lo que devuelven ahora:

```
{  
  "notificationId": "COMM-EXEC-20241209-001234",  
  "status": "SENT",  
  "submittedAt": "2024-12-09T15:30:25.123Z",  
  "channel": "SMS",  
  "communicationExecutionId": "BIAN-COMM-EXEC-UUID-12345"  
}
```

Lo que necesitamos (para cumplir Epic 9 - Analytics):

```
{  
  "notificationId": "COMM-EXEC-20241209-001234",  
  "status": "SENT",  
  "submittedAt": "2024-12-09T15:30:25.123Z",  
  "channel": "SMS",  
  "communicationExecutionId": "BIAN-COMM-EXEC-UUID-12345",  
  
  // ⭐ METADATA CRÍTICA (SOLICITAR)  
  "providerMetadata": {  
    "actualProvider": "TWILIO_US",  
    "providerLatencyMs": 95,  
    "mulesoftLatencyMs": 25,  
    "totalLatencyMs": 120,  
    "providerCost": 0.05,  
    "fallbackUsed": false  
  }  
}
```

PREGUNTAS TÉCNICAS DETALLADAS

1. Documentación API

 API Encontrada: Singular Bank - Communication Services API (ID: 2611145)

Acceso a Documentación:

- Anypoint Exchange URL completa:** ¿Cuál es el link directo para compartir con el equipo?
 - Implementation URI completa:** <https://api.selfbank.es/system/commu...> (está truncada, necesito la URL completa)
 - Especificación RAML 1.0:** ¿Puedo descargar el archivo completo?
 - OpenAPI 3.0:** ¿Está disponible en formato OpenAPI/Swagger?
 - Postman Collection:** ¿Tienen collection generada para importar?

Conformidad de la API:

- **⚠ Instance Conformance: Not Validated** → ¿Los responses reales cumplen 100% con el schema RAML?
 - ¿Hay diferencias conocidas entre la especificación y la implementación real?
 - ¿Están planeando validar la conformidad?

2. Endpoints Disponibles CONFIRMADOS

Información obtenida de la documentación:

 SMS (DISPONIBLE):

 **Endpoint:** POST /communication-execution/sms-notification/execute

 **Request schema:** Ver ejemplo completo en documentación

```
{  
  "customerId": "CUST12345678",  
  "practice": "monkey", // ? PREGUNTAR QUÉ ES ESTO  
  "channel": "SMS",  
  "recipient": {  
    "phoneNumber": "+34653093774",  
    "countryCode": "ES"  
  },  
  "content": {  
    "message": "Texto del mensaje",  
    "encoding": "UTF8"  
  },  
  "smsOptions": {
```

```

        "senderId": "SELFBANK",
        "validityPeriod": 60,
        "deliveryReport": true
    },
    "metadata": {
        "campaignId": "...",
        "correlationId": ...
    }
}

```

Response schema:

```
{
    "notificationId": "COMM-EXEC-20241209-001234",
    "status": "SENT",
    "submittedAt": "2024-12-09T15:30:25.123Z",
    "channel": "SMS",
    "communicationExecutionId": "BIAN-COMM-EXEC-UUID-12345"
}
```

- Provider subyacente:** ¿Usan Twilio, Nexmo, otro? → **PREGUNTAR**
- Encoding:** ¿Soportan emojis? ¿Límite de caracteres?
- SenderId:** ¿"SELFBANK" es el único permitido o puedo personalizarlo?

PUSH (DISPONIBLE):

- Endpoint:** POST /communication-execution/push-notification/execute
- Request schema:** ¿Igual que SMS? ¿Campos específicos para Push?
- deviceToken:** ¿Cómo se especifica?
- Provider subyacente:** ¿Firebase FCM, otro? → **PREGUNTAR**

EMAIL (DISPONIBLE - BONUS):

- Endpoint:** POST /communication-execution/email-notification/execute
- Provider:** Microsoft Outlook 365 (Graph API)
- Autenticación:** ¿Necesitamos configurar algo en nuestro lado?

VOICE (NO DISPONIBLE - FUERA DE ALCANCE):

- DECISIÓN:** No se implementará en Epic 11
- Solo informativo: ¿Está en roadmap de MuleSoft para futuro?

BIOMETRIC (NO DISPONIBLE - FUERA DE ALCANCE):

- DECISIÓN:** No se implementará en Epic 11
- Solo informativo: ¿Está en roadmap de MuleSoft para futuro?

MONITOREO (DISPONIBLE):

- Health Check:** GET /health/retrieve
- Métricas:** GET /metrics/retrieve
- Pregunta:** ¿Qué información devuelven estos endpoints?
 - ¿Estado por provider? (Twilio UP/DOWN, Firebase UP/DOWN)
 - ¿Latencias por provider?
 - ¿Métricas de uso?

3. Autenticación & Seguridad

¿Cómo me autentico con MuleSoft API?

- Método:** ¿OAuth2, API Key, mTLS, JWT Bearer?
- Credenciales DEV:** ¿Cómo las obtengo?
- Credenciales UAT:** ¿Cómo las obtengo?
- Credenciales PROD:** ¿Cómo las obtengo?

Si es **OAuth2**:

- ¿Cuál es el **token endpoint**?
- ¿Qué **grant type** usan? (client_credentials, authorization_code)
- ¿Los tokens **expiran**? ¿Cada cuánto?
- ¿Cómo **renuevo** el token?
- ¿Qué **scopes** necesito solicitar?

Si es **API Key**:

- ¿Dónde va la key? (header, query param)
- ¿Qué nombre tiene el header? (ej: X-API-Key, Authorization)

Otros:

- ¿Necesito **IP whitelisting**? ¿Qué IPs debo registrar?
- ¿Hay **certificados TLS/mTLS** requeridos?

4. Ambientes & URLs

¿Qué URLs uso para cada ambiente?

DEV/Sandbox: `https://???`

UAT/Staging: `https://???`

PROD: `https://???`

Conectividad:

¿Necesito **VPN** para acceder?

¿Necesito **firewall rules** configuradas?

¿Los **endpoints son iguales** en todos los ambientes? (solo cambia base URL)

5. SLAs, Timeouts & Rate Limits

¿Cuáles son los límites y garantías de servicio?

Timeouts:

¿Cuál es el **timeout máximo recomendado** por request?

¿Cuál es el **tiempo de respuesta promedio**? (P50, P95, P99)

Rate Limits:

¿Cuántos **requests por segundo** puedo enviar?

¿Es por **IP**, por **API key**, o por **aplicación**?

¿Qué pasa si excedo el límite? → HTTP 429 + `Retry-After` header?

SLAs:

Disponibilidad garantizada: 99.9%? 99.5%?

Latency garantizada: P99 < 500ms? P95 < 300ms?

Retry Policy:

¿Cuántos **reintentos** permiten antes de bloquear?

¿Qué **backoff strategy** recomiendan? (exponencial, lineal)

6. ● Manejo de Errores

¿Qué códigos de error devuelven y qué significan?

- 400 Bad Request:** ¿Qué errores específicos? (validación, campos faltantes)
- 401 Unauthorized:** ¿Token inválido o expirado?
- 403 Forbidden:** ¿Falta de permisos?
- 429 Too Many Requests:** ¿Rate limit excedido? → Retry-After header?
- 500 Internal Server Error:** ¿Error de MuleSoft?
- 503 Service Unavailable:** ¿Provider caído?

Formato de errores:

- ¿Tienen **formato estandarizado** de error response?

```
{  
  "code": "INVALID_PHONE",  
  "message": "Phone number format invalid",  
  "details": { ... },  
  "timestamp": "2025-11-30T10:00:00Z"  
}
```

7. 🚗 Testing & Sandbox

¿Cómo puedo probar sin consumir providers reales?

- ¿Tienen **sandbox environment** con **mock providers**?
- ¿Puedo hacer requests de prueba **sin costo** en DEV?
- ¿Tienen **datos de prueba** (teléfonos, deviceTokens) que pueda usar?
- ¿Requieren que hagamos **contract tests**? (Pact, Spring Cloud Contract)
- ¿Tienen **mock server** o **WireMock stubs** disponibles?

8. 📈 Monitoreo & Observabilidad

¿Cómo monitoreo el estado y uso de la API?

- ¿Tienen **dashboards de MuleSoft** que pueda ver?
- ¿Proveen **métricas** de mi consumo? (requests, latency, errores)
- ¿Dónde puedo ver **logs** de mis requests?
- ¿Tienen **alertas** configuradas? ¿Cómo me notifican?
- ¿Cómo reporto **incidentes**? (Slack, Jira, email)

- ¿Hay **status page** para ver disponibilidad de MuleSoft?
-

9. Configuración de Providers (en MuleSoft)

¿Cómo funciona la selección de provider en el lado de MuleSoft?

Preguntas:

- ¿MuleSoft elige el provider automáticamente? (transparente para nosotros)
- ¿O debemos **especificar qué provider usar** en el request? (header, query param, body field)
- ¿MuleSoft maneja **fallback automático** si un provider falla?
- ¿MuleSoft tiene **múltiples providers** por canal? (ej: Twilio + Nexmo para SMS)
- ¿Podemos **configurar preferencias** de provider en MuleSoft?

 **Nota:** Signature Router **NO** se comunicará directamente con providers (Twilio, Firebase). Solo con MuleSoft.

9.1. CRÍTICO: Observabilidad y Metadata de Providers NUEVO

Contexto: Signature Router tiene un dashboard de métricas y analítica (Epic 9) que muestra:

- Performance por proveedor real (Twilio, Firebase, Vonage, etc.)
- Latencia específica de cada proveedor
- Tasa de éxito/error por proveedor
- Costos por proveedor (si disponible)
- A/B testing entre providers
- Alertas de degradación de provider específico

PROBLEMA: Si MuleSoft actúa como gateway opaco, **perdemos toda esta visibilidad**.

Requerimiento CRÍTICO: Metadata Enriquecida en Responses

¿MuleSoft puede incluir metadata del provider real en cada response?

Ejemplo del response que necesitamos:

```
{  
  "transactionId": "SM123abc456",  
  "status": "success",  
  "message": "SMS sent successfully",
```

```

// ⭐ METADATA CRÍTICA (esto es lo que NECESITAMOS)

"metadata": {
    "actualProvider": "TWILIO_US",           // ← Provider real usado
    "providerType": "SMS",                  // ← Tipo de canal
    "providerLatencyMs": 95,                // ← Latencia solo del provider
    externo
    "mulesoftLatencyMs": 25,                // ← Overhead de MuleSoft (opcional
    pero útil)
    "totalLatencyMs": 120,                  // ← Latencia total
    "providerCost": 0.05,                   // ← Coste del envío (si disponible)
    "fallbackUsed": false,                 // ← Si hubo fallback interno en
    MuleSoft
    "attemptNumber": 1,                    // ← Número de intento
    "timestamp": "2025-11-30T23:00:00Z"
}
}

```

Preguntas Específicas:

- ¿Los interfaces actuales de MuleSoft ya incluyen esta metadata?
- Si NO, ¿es posible **ampliar los interfaces** para incluirla?
 - **Nota:** Entendemos que los interfaces **NO son inmutables** y pueden modificarse.
 - Si necesitamos esta funcionalidad, **la vamos a solicitar formalmente.**
- ¿Qué información del provider real pueden exponer?
 - Nombre/ID del provider usado (ej: "TWILIO_US", "FIREBASE_FCM")
 - Latencia específica del provider (sin incluir overhead de MuleSoft)
 - Código de error original del provider (si falló)
 - Mensaje de error original del provider
 - Coste del envío (si MuleSoft tiene esta información)
 - Si hubo fallback automático en el lado de MuleSoft

Errores Específicos del Provider

Cuando un envío **falla**, necesitamos saber:

```

{
  "status": "error",
  "errorCode": "PROVIDER_ERROR",           // ← Código general
  "errorMessage": "Failed to send SMS",

  // ⭐ METADATA DE ERROR (CRÍTICA para troubleshooting)
  "metadata": {

```

```

    "errorSource": "TWILIO_US",           // ← Qué provider falló
    "providerErrorCode": "21211",          // ← Código original de Twilio
    "providerErrorMessage": "The 'To' number +341234 is not a valid phone
number.",
    "providerResponseTime": 45,
    "attemptsMade": 2,
    "fallbackAttempted": true,
    "fallbackProvider": "VONAGE_EU",
    "fallbackResult": "ALSO_FAILED"
}
}

```

Impacto en Funcionalidades si NO hay Metadata:

Funcionalidad	Con Metadata	Sin Metadata	Impacto
Dashboard de Providers	✓ Twilio: 95%, Firebase: 88%	✗ Solo "MuleSoft": 90%	⚠ ALTO – Pérdida de visibilidad
Alertas por Provider	✓ "Twilio degradado (80%)"	✗ "MuleSoft degradado"	⚠ ALTO – Alertas genéricas
Optimización de Costos	✓ Comparar costos por provider	✗ Coste total opaco	⚠ MEDIO – No hay optimización
A/B Testing	✓ 50% Twilio, 50% Vonage	✗ No posible	⚠ ALTO – No hay testing
Troubleshooting	✓ "Twilio error: Invalid phone"	✗ "MuleSoft error genérico"	⚠ CRÍTICO – Dificulta debugging
SLA Tracking	✓ Latencia por provider	✗ Latencia total (MuleSoft + Provider)	⚠ MEDIO – SLAs imprecisos

Propuesta de Solución:

Si los interfaces actuales **NO incluyen** esta metadata:

- 1. Opción A (PREFERIDA):** Ampliar los interfaces de MuleSoft para incluir campo `metadata` en responses
 - Backward compatible (campo opcional)
 - Habilitado por feature flag o header (ej: `X-Include-Provider-Metadata: true`)
 - Implementación incremental por canal

2. Opción B: Headers HTTP con metadata

```
X-Provider-Name: TWILIO_US  
X-Provider-Latency: 95  
X-Provider-Cost: 0.05
```

3. Opción C: Endpoint separado para métricas

```
GET /api/v1/transactions/{transactionId}/metrics
```

Health Check de Providers

¿MuleSoft expone el **estado de salud de cada provider** subyacente?

Endpoint ideal:

```
GET /api/v1/providers/health

Response:
{
  "overall": "HEALTHY",
  "providers": [
    {
      "providerId": "TWILIO_US",
      "type": "SMS",
      "status": "UP",
      "latencyMs": 45,
      "successRate": 0.95,
      "lastCheckAt": "2025-11-30T23:00:00Z"
    },
    {
      "providerId": "FIREBASE_FCM",
      "type": "PUSH",
      "status": "DOWN",
      "error": "Authentication failed",
      "lastCheckAt": "2025-11-30T23:00:00Z"
    }
  ]
}
```

10. Resiliencia (Circuit Breaker, Retry)

¿Quién maneja los patrones de resiliencia?

Signature Router tiene capacidad para:

- Circuit Breaker (Resilience4j)

- Retry con exponential backoff
- Timeout configuration

Preguntas sobre división de responsabilidades:

- ¿MuleSoft tiene **circuit breaker** implementado hacia los providers?
- ¿MuleSoft hace **retry automático** si un provider externo falla?
- ¿Esperan que nosotros implementemos **circuit breaker/retry** hacia MuleSoft?
- ¿O MuleSoft garantiza disponibilidad y nosotros solo manejamos timeouts?

Coordinación:

- ¿Cómo se **coordina** la resiliencia entre ambos sistemas para evitar "retry storms"?
- ¿Cómo notifican **degradación de servicio**? (header X-Service-Status, status code)
- Si un provider externo está caído, ¿MuleSoft devuelve **503** inmediatamente o intenta retry?

11. Migración & Rollout

¿Cuál es el plan recomendado para la migración?

- ¿Cuándo podemos **empezar a integrar**? (timeline de acceso)
- ¿Cuál es el **proceso de onboarding**?
- ¿Recomiendan **canary deployment**? (10% → 25% → 50% → 100%)
- ¿Hay alguna **ventana de mantenimiento** que deba considerar?

Estrategia de migración por canal:

- ¿Puedo migrar **un canal a la vez**? (ej: SMS primero, luego PUSH)
- ¿O debo migrar **todos los canales simultáneamente**?
- ¿Qué **orden** recomiendan? (SMS → PUSH → VOICE → BIOMETRIC)

Rollback:

- Si hay problemas con MuleSoft, ¿puedo hacer **rollback temporal** a providers directos?
- ¿O la normativa **prohíbe estrictamente** comunicación directa (sin excepciones)?
- ¿Qué hacer en caso de **incidente crítico** de MuleSoft en producción?

12. Soporte & Contactos

¿A quién contacto si hay problemas?

- ¿Quién es el **contacto técnico principal**?
 - ¿Tienen **Slack channel** de soporte?
 - ¿Cuál es el **SLA de respuesta** a issues?
 - ¿Hay **oncall/guardia** 24/7 para producción?
 - ¿Cómo escalo **incidentes críticos**?
-

13. Request/Response Examples

¿Pueden darme ejemplos reales de request/response?

Necesito ver:

- Request completo** con headers, auth, body para SMS
- Response exitoso** (200 OK)
- Response con error** (400, 429, 500, 503)

Ejemplo ideal:

```
# Request
curl -X POST https://mulesoft.company.com/api/v1/sms \
-H "Authorization: Bearer TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "recipient": {
    "phoneNumber": "+34612345678"
  },
  "message": {
    "body": "Your code is: 123456"
  }
}'

# Response 200 OK
{
  "messageId": "SM1234567890",
  "status": "SENT",
  "timestamp": "2025-11-30T10:00:00Z"
}
```

14. 💰 Costos & Billing

¿Hay algún costo asociado al uso de MuleSoft API?

- ¿Hay límite de requests gratis?
 - ¿Cómo se facturan los requests?
 - ¿Hay alertas cuando me acerco a límites de presupuesto?
 - ¿Puedo ver consumo en tiempo real?
-

15. 🔎 Casos Especiales & Edge Cases

Situaciones específicas que debo considerar:

Fallos de providers externos:

- ¿Qué pasa si provider externo está caído? (Twilio down, Firebase down)
- ¿MuleSoft devuelve 503 inmediatamente o intenta con provider secundario?
- ¿Tienen redundancia de providers? (ej: Twilio primario + Nexmo secundario)
- ¿Cómo me notifican que un provider específico está degradado?

Validaciones:

- ¿Cómo manejan mensajes largos (SMS > 160 chars)? ¿Segmentación automática?
- ¿Soportan números internacionales? ¿Todos los países?
- ¿Hay validación de formato de phoneNumber en su lado?
- ¿Qué validaciones hacen antes de llamar al provider? (formato, blacklist, etc.)

Idempotencia:

- ¿Cómo manejan duplicados? (enviar mismo SMS 2 veces)
- ¿Aceptan idempotency key en headers? (ej: X-Idempotency-Key)
- ¿Durante cuánto tiempo cachean requests duplicados? (5 min, 1 hora)

Canales no disponibles:

- Si un canal NO está implementado en MuleSoft, ¿qué código de error devuelven?
 - ¿Devuelven 501 Not Implemented o 404 Not Found?
 - ¿El error response indica qué canales SÍ están disponibles?
-

Entregables que Necesito

Al final de la reunión, por favor solicita:

1. **Lista de canales disponibles CONFIRMADO:** SMS, PUSH, EMAIL
 VOICE y BIOMETRIC: Fuera de alcance (no se implementarán)
2. **Documentación OBtenida:** Tengo PDF de Exchange
 - Solicitar:** Archivo RAML completo descargable
 - Solicitar:** OpenAPI 3.0 (si disponible)
3. **Postman Collection - PENDIENTE:** Solicitar collection con ejemplos
4. **Credenciales de DEV/Sandbox - CRÍTICO:** Client ID + Client Secret
 - DEV environment
 - UAT environment
 - PROD environment (para futura migración)
5. **URLs completas de ambientes - PENDIENTE:**
 - DEV: [https://????/communication-execution/...](https://????/communication-execution/)
 - UAT: [https://????/communication-execution/...](https://????/communication-execution/)
 - PROD: [https://????/communication-execution/...](https://????/communication-execution/)
6. **Contacto técnico principal** (email, Slack, Teams) - PENDIENTE
7. **Explicación del campo "practice": "monkey"** - CRÍTICO
8. **Providers reales usados - CRÍTICO:**
 - SMS: ¿Twilio? ¿Nexmo? ¿Otro?
 - PUSH: ¿Firebase? ¿OneSignal? ¿Otro?
9. **Timeline de migración** (cuándo podemos empezar, cuándo go-live) - PENDIENTE
10. **Espectificaciones de endpoints /health y /metrics - NUEVO:**
 - ¿Qué información devuelve /health/retrieve?
 - ¿Qué métricas incluye /metrics/retrieve?
 - ¿Incluyen info por provider?

Checklist Post-Reunión

Después de la reunión, validar que tengo:

- Lista clara de canales disponibles** (cuáles SÍ, cuáles NO)
- OpenAPI spec descargada** para cada canal disponible

- Credenciales almacenadas en Vault (DEV, UAT, PROD)
 - URLs de ambientes documentadas
 - SLAs y rate limits claros (timeouts, requests/seg, latency)
 - Contacto técnico agregado a Slack/Email
 - Timeline de migración acordado (fecha inicio, fecha go-live)
 - Acceso a sandbox funcionando
 - Primer request de prueba exitoso (al menos 1 canal)
 - Entendimiento claro de canales NO disponibles y cómo impactan al proyecto
-

Próximos Pasos Post-Reunión

Una vez tengas toda esta información:

1. Documentar Canales Disponibles

Crear: docs/architecture/mulesoft-canales-disponibles.md

```
# Canales Disponibles en MuleSoft

## ✅ Canales Implementados
- SMS: Disponible (endpoint: /api/v1/sms)
- PUSH: Disponible (endpoint: /api/v1/push)

## ❌ Canales NO Disponibles
- VOICE: No planificado
- BIOMETRIC: Roadmap Q2 2026

## 📊 Impacto en Signature Router
- Funcionalidades disponibles: SMS, PUSH
- Funcionalidades NO disponibles: VOICE, BIOMETRIC
```

2. Guardar Especificaciones

- Guardar OpenAPI spec en: docs/architecture/mulesoft-api-spec.yaml
- Documentar hallazgos en: docs/architecture/mulesoft-api-reference.md
- Actualizar: docs/architecture/08-mulesoft-integration-strategy.md

3. Actualizar PRD y Epics

- **Si canales NO disponibles:** Remover/postponer epics afectadas
- **Si todos disponibles:** Proceder con Epic 11 completa
- Actualizar `docs/epics.md` según canales reales

4. Generar Epic 11

- Avisar al equipo para generar **Epic 11** con specs reales
- Incluir **SOLO** los canales que MuleSoft tiene disponibles
- Agregar stories para canales futuros en backlog (si aplica)

Preparado para: Reunión MuleSoft - Lunes

Objetivo: Obtener 100% de información técnica para implementar Epic 11

Resultado esperado: Poder empezar desarrollo en 1-2 días post-reunión

Tips para la Reunión

1. **Graba la reunión** (si es virtual) para no perder detalles
2. **Empieza preguntando por canales disponibles** – es la pregunta MÁS importante
3. **Toma notas** de URLs, nombres técnicos, procesos
4. **Pide que compartan pantalla** cuando muestren documentación
5. **Solicita acceso inmediato** a todos los recursos (no esperar días)
6. **Confirma timeline realista** de cuándo puedes empezar a integrar
7. **Pregunta por casos límite** (qué hacer cuando hay problemas, provider caído, etc.)
8. **Clarifica normativa** de comunicación directa (¿permitido en emergencias?)
9. **Verifica que tengas todo** antes de terminar la reunión

ESCENARIOS ESPERADOS

Escenario CONFIRMADO: SMS + PUSH disponibles

Impacto: Signature Router soportará SMS y PUSH

Acción: Epic 11 con SMS y PUSH únicamente

Canales fuera de alcance: VOICE y BIOMETRIC no se implementarán

Escenarios alternativos (DESCARTADOS)

- Escenario A: Solo SMS
 - Escenario C: SMS + PUSH + VOICE
 - Escenario D: Todos los canales
-

Objetivo de la Reunión

Salir con claridad absoluta de:

1. ~~Qué canales SÍ puedo implementar~~ **CONFIRMADO:** SMS + PUSH
 2. ~~Qué canales NO están disponibles~~ **CONFIRMADO:** VOICE + BIOMETRIC (fuera de alcance)
 3. **CRÍTICO:** Obtener credenciales y URLs para empezar desarrollo
 4. **CRÍTICO:** Aclarar campo "practice": "monkey"
 5. **CRÍTICO:** Schema completo de PUSH
 6.  Timeline realista para **empezar desarrollo**
-

ANEXO: Justificación de Negocio para Metadatos de Providers

¿Por qué es CRÍTICO tener visibilidad del provider real?

1. Cumplimiento de SLAs Contractuales

- Nuestro contrato con clientes **garantiza P99 < 500ms** para envío de SMS
- Si MuleSoft + Twilio tardan 600ms, necesitamos saber:
 - ¿Es Twilio lento (500ms)? → Cambiar de provider
 - ¿Es MuleSoft lento (400ms)? → Optimizar integración
- **Sin metadata, no podemos optimizar ni cumplir SLAs**

2. Optimización de Costos

- Twilio US: \$0.05 por SMS
- Vonage EU: \$0.03 por SMS
- **Ahorro potencial: 40%** si cambiamos de provider para ciertos destinos
- **Sin metadata de costos, perdemos oportunidad de ahorro**

3. Troubleshooting y Soporte

- Cliente reporta: "No recibí el SMS"
- **Con metadata:** "Twilio error 21211: Número inválido" → Solución inmediata
- **Sin metadata:** "Error de MuleSoft" → Escalamos a MuleSoft → Ellos escalan a Twilio → **+24h de resolución**
- **Impacto:** Experiencia de usuario degradada, SLA de soporte incumplido

4. Capacidad de Reacción ante Incidentes

- Si Twilio tiene un outage regional (ocurre ~2 veces/año)
- **Con metadata:** Detectamos inmediatamente "Twilio US: 0% success" → Activamos fallback a Vonage
- **Sin metadata:** Solo vemos "MuleSoft: 50% success" → No sabemos qué provider tiene problema
- **Impacto:** Tiempo de reacción 10x más lento (minutos vs. segundos)

5. Mejora Continua y A/B Testing

- Queremos probar: "¿Vonage es más rápido que Twilio para destinos EU?"
- **Con metadata:** A/B test con 50% tráfico a cada uno → Medimos latencias reales
- **Sin metadata:** Imposible hacer testing → Quedamos con provider subóptimo
- **Impacto:** No podemos mejorar performance ni costos

6. Auditoría y Compliance

- Regulación bancaria requiere trazabilidad completa de transacciones críticas
- **Con metadata:** "SMS enviado vía Twilio US, SID: SM123, latencia: 95ms, coste: \$0.05"
- **Sin metadata:** "SMS enviado vía MuleSoft" (insuficiente para auditoría)
- **Impacto:** Posible incumplimiento regulatorio

7. Análisis Predictivo y ML

- Queremos predecir: "¿Qué provider tendrá mejor tasa de entrega para este destino?"
- **Requiere:** Datos históricos granulares por provider
- **Sin metadata:** Dataset incompleto → Modelos de ML imposibles
- **Impacto:** No podemos implementar smart routing predictivo

Contraargumento Esperado de MuleSoft

"Ustedes no deberían preocuparse por los providers internos, ese es nuestro problema"

Nuestra Respuesta:

- ✓ **Entendemos y valoramos** que MuleSoft gestione la complejidad de providers
- ✓ **No queremos** gestionar credenciales, certificados, ni integraciones directas
- ✓ **Pero SÍ necesitamos** visibilidad para cumplir **nuestros SLAs con clientes finales**

La metadata NO rompe la abstracción, solo la hace **observable**.

Analogía: Un CDN (Cloudflare, Akamai) abstrae la complejidad de edge servers, pero **SÍ expone** qué datacenter sirvió cada request (header `X-Edge-Location`). Esto permite optimizaciones sin romper la abstracción.

Propuesta WIN-WIN 🤝

Para MuleSoft:

- ✓ Siguen siendo **la única** capa de integración (cumple normativa)
- ✓ No tienen que cambiar lógica interna, solo **exponer metadata existente**
- ✓ Mejora la **calidad de servicio percibida** por clientes internos
- ✓ Reduce escalaciones de soporte (nosotros debugging más rápido)

Para Signature Router:

- ✓ Mantenemos **observabilidad completa** del sistema
- ✓ Cumplimos **SLAs contractuales** con clientes
- ✓ Optimizamos **costos** basados en datos reales
- ✓ Troubleshooting **10x más rápido**

Implementación Sugerida (Mínima Fricción)

Fase 1: Metadata Básica (MVP) - 1 sprint

```
{  
  "transactionId": "...",  
  "status": "success",  
  "metadata": {  
    "actualProvider": "TWILIO_US",      // Solo el nombre  
    "providerLatencyMs": 95           // Solo la latencia
```

```
}
```

Fase 2: Metadata Completa - 2 sprints

```
{
  "metadata": {
    "actualProvider": "TWILIO_US",
    "providerLatencyMs": 95,
    "mulesoftLatencyMs": 25,
    "providerCost": 0.05,
    "fallbackUsed": false
  }
}
```

Fase 3: Health Endpoint - 1 sprint

```
GET /api/v1/providers/health
```

Timeline total: 4 sprints (~2 meses)

Esfuerzo estimado: Bajo (metadata ya existe internamente, solo exponer)

¡Buena suerte en la reunión! 

 **Recuerda:**

1. La pregunta de canales disponibles define TODO el alcance de Epic 11. ¡Es la MÁS importante!
2. La metadata de providers es CRÍTICA para observabilidad. **No es negociable si queremos un sistema production-ready.**